AF $

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

re application of	:	**Confirmation No. 5407**
Akira KAMIYA	:	Attorney Docket No. 2001_0976A
Serial No. 09/900,168	:	Group Art Unit 2621
Filed July 9, 2001	:	Examiner Richard J. Lee
APPARATUS AND METHOD OF MULTIPLE DECODING	:	**Mail Stop: Appeal Brief - Patents**

PATENT OFFICE FEE TRANSMITTAL FORM

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

THE COMMISIONER IS AUTHORIZED TO CHARGE ANY DEFICIENCY IN THE FEES FOR THIS PAPER TO DEPOSIT ACCOUNT NO. 23-0975

Sir:

Attached hereto is a check in the amount of $500.00 to cover Patent Office fees relating to filing the following attached papers:

Appeal Brief $500.00

A duplicate copy of this paper is being submitted for use in the Accounting Division, Office of Finance.

The Commissioner is authorized to charge any deficiency or to credit any overpayment associated with this communication to Deposit Account No. 23-0975, with the EXCEPTION of deficiencies in fees for multiple dependent claims in new applications.

Respectfully submitted,

Akira KAMIYA

By _____
David M. Ovedovitz
Registration No. 45,336
Attorney for Applicant

DMO/jmj
WENDEROTH, LIND & PONACK, L.L.P.
2033 K St., N.W., Suite 800
Washington, D.C. 20006-1021
Telephone (202) 721-8200
March 20, 2007

[Check No. 79466]

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	**Confirmation No. 5407**
Akira KAMIYA	:	Attorney Docket No. 2001_0976A
Serial No. 09/900,168	:	Group Art Unit 2621
Filed July 9, 2001	:	Examiner Richard J. Lee
APPARATUS AND METHOD OF MULTIPLE DECODING	:	**Mail Stop: Appeal Brief - Patents**

APPELLANT'S BRIEF UNDER 37 CFR § 41.37

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The following is an Appellant's Brief for the above-identified U.S. patent application, submitted in accordance with the provisions of 37 CFR 41.37.

Real Party in Interest

The real party in interest is Matsushita Electric Industrial Co., Ltd of Osaka Japan, the Assignee of the above-referenced application.

Related Appeals and Interferences

There are no known related appeals or interferences.

Status of Claims

Claims 1, 2, 5 and 6 have been canceled. Claims 3, 4, 7 and 8 are presently pending. Pending claims 3, 4, 7 and 8 were finally rejected in the Office Action of October 24, 2006, and the rejection of these claims is appealed. A complete copy of the claims on appeal is provided in Appendix I.

Status of Amendments

No amendments subsequent to the final rejection of October 24, 2006 have been made.

Summary of the Claimed Subject Matter

A description of the subject matter recited in the rejected claims will now be provided below with reference to the written description and the drawings of this application.

Claim 3 recites a multiple decoding apparatus for receiving a broadcasting signal composed of a plurality of encoded data and for simultaneously decoding two or more of the encoded data (*see, for example,* paragraphs [0001] and [0008] and Figure 1), the multiple decoding apparatus comprising:

a reproduction controller for outputting control information related to decoding and reproduction of data (*see, for example,* paragraph [0027] and Figure 1, element 190);

a data extractor for receiving the broadcasting signal and extracting at least audio data and video data which are designated by the control information (*see, for example,* paragraph [0027] and Figure 1, element 110);

a buffer for storing at least the audio data and the video data extracted by said data extractor (*see, for example,* paragraph [0027] and Figure 1, element 120);

a buffer manager for controlling said buffer in accordance with the control information for said buffer (*see, for example,* paragraphs [0027] and [0030] and Figure 1, element 160);

a data flow controller for distributing at least the audio data and the video data stored in said buffer for each data type and transferring at least the audio data and the video data in accordance with provided transfer conditions (*see, for example,* paragraphs [0027] and [0032] and Figure 1, element 130);

a plurality of separate buffers for respectively storing at least the audio data and the video data distributed and transferred by said data flow controller according to each data type (*see, for example,* paragraph [0027] and Figure 1, elements 141-14n);

a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in said plurality of separate buffers so as to be associated with each other in accordance with information for specifying said plurality of separate buffers (*see, for example,* paragraphs [0027] and [0033] and Figure 1, element 170);

a plurality of decoders respectively corresponding to said plurality of separate buffers for decoding at least the audio data and the video data stored in said plurality of separate buffers and outputting two or more decoded data (*see, for example,* paragraph [0027] and [0029] and Figure 1 elements 151-15n); and

a decoding controller for selecting a separate buffer and a decoder, which are used for the decoding, according to a usage status of said decoder from among said plurality of separate buffers and said plurality of decoders in accordance with the control information, and outputting information related to said separate buffer selected by said decoding controller, the transfer conditions based on said separate buffer selected by said decoding controller, and an instruction to start decoding, respectively, to said separate buffer manager, said data flow controller, and said decoder selected by said decoding controller (*see, for example,* paragraphs [0027] and [0030] and Figure 1, element 180), wherein

said separate buffer manager outputs, when a specific separate buffer becomes full of data, an overflow notification that said specific separate buffer overflows to said decoding controller (*see, for example,* paragraphs [0034] – [0040], Figure 3, step S307, and Figure 5),

said decoding controller outputs, upon receipt of the overflow notification that said specific separate buffer overflows, an instruction to stop data transfer to said specific separate buffer to said data flow controller, outputs an instruction to stop decoding to a decoder corresponding to said specific separate buffer, and outputs an instruction to initialize said specific separate buffer to said separate buffer manager (*see, for example,* paragraphs [0037] - [0039] and Figure 5),

said separate buffer manager initializes said specific separate buffer in accordance with the instruction to initialize said specific separate buffer from said decoding controller without initializing said buffer (*see, for example,* paragraphs [0037] - [0039] and Figure 5), and

the multiple decoding apparatus resumes all processing which was stopped as a result of said specific separate buffer becoming full after said specific separate buffer is initialized (*see, for example,* paragraphs [0037] - [0039] and Figure 5).

Claim 4 recites a multiple decoding apparatus for receiving a broadcasting signal composed of a plurality of encoded data and for simultaneously decoding two or more of the

encoded data (*see, for example,* paragraphs [0001] and [0008] and Figure 1), the multiple decoding apparatus comprising:

a reproduction controller for outputting control information related to decoding and reproduction of data (*see, for example,* paragraph [0027] and Figure 1, element 190);

a data extractor for receiving the broadcasting signal and extracting at least audio data and video data which are designated by the control information (*see, for example,* paragraph [0027] and Figure 1, element 110);

a buffer for storing at least the audio data and the video data extracted by said data extractor (*see, for example,* paragraph [0027] and Figure 1, element 120);

a buffer manager for controlling said buffer in accordance with the control information for said buffer (*see, for example,* paragraphs [0027] and [0030] and Figure 1, element 160);

a data flow controller for distributing at least the audio data and the video data stored in said buffer for each data type and transferring at least the audio data and the video data in accordance with provided transfer conditions (*see, for example,* paragraphs [0027] and [0032] and Figure 1, element 130);

a plurality of separate buffers for respectively storing at least the audio data and the video data distributed and transferred by said data flow controller according to each data type (*see, for example,* paragraph [0027] and Figure 1, elements 141-14n);

a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in said plurality of separate buffers so as to be associated with each other in accordance with information for specifying said plurality of separate buffers (*see, for example,* paragraph (*see, for example,* paragraphs [0027] and [0033] and Figure 1, element 170);

a plurality of decoders respectively corresponding to said plurality of separate buffers for decoding at least the audio data and the video data stored in said plurality of separate buffers and outputting two or more decoded data (*see, for example,* paragraph [0027] and [0029] and Figure 1 elements 151-15n); and

a decoding controller for selecting a separate buffer and a decoder, which are used for the decoding, according to a usage status of said decoder from among said plurality of separate buffers and said plurality of decoders in accordance with the control information, and outputting information related to said separate buffer selected by said decoding controller, the transfer conditions based on said separate buffer selected by said decoding controller, and an instruction

to start decoding, respectively, to said separate buffer manager, said data flow controller, and said decoder selected by said decoding controller (*see, for example,* paragraphs [0027] and [0030] and Figure 1, element 180), wherein

said separate buffer manager outputs, when a specific separate buffer becomes full of data, an overflow notification that said specific separate buffer overflows to said decoding controller (*see, for example,* paragraphs [0034] – [0040], Figure 3, step S307, and Figure 6),

said decoding controller outputs, upon receipt of the overflow notification that said specific separate buffer overflows, an instruction to discard encoded data directed toward said specific separate buffer to said data flow controller, outputs an instruction to stop decoding to a decoder corresponding to said specific separate buffer, and outputs an instruction to initialize said specific separate buffer to said separate buffer manager (*see, for example,* paragraphs [0040] and [0041] and Figure 6),

said separate buffer manager initializes said specific separate buffer in accordance with the instruction to initialize said specific separate buffer from said decoding controller without initializing said buffer (*see, for example,* paragraphs [0040] and [0041] and Figure 6), and

the multiple decoding apparatus resumes all processing which was stopped as a result of said specific separate buffer becoming full, and the discard of the encoded data is released after said specific separate buffer is initialized (*see, for example,* paragraphs [0040] and [0041] and Figure 6).

Claim 7 recites a multiple decoding method for simultaneously decoding two or more encoded data from a broadcasting signal composed of a plurality of encoded data (*see, for example,* paragraphs [0001] and [0008] and Figure 1), the multiple decoding method comprising:

selecting a plurality of decoders for performing decoding and a plurality of separate buffers corresponding to the plurality of decoders, respectively, according to usage status of the plurality of decoders (*see, for example,* paragraphs [0029] and [0030] and Figure 2, steps 202-205);

extracting at least audio data and video data to be decoded and reproduced from the broadcasting signal (*see, for example,* paragraph 30 and Figure 2 steps 209 and 210);

storing at least the extracted audio data and video data in a buffer (*see, for example,* paragraph [0031] and Figure 3, step 303);

distributing at least the audio data and the video data stored in the buffer for each data type and respectively storing at least the audio data and the video data in the plurality of separate buffers according to each data type (*see, for example,* paragraph [0032] and Figure 3, steps 304 and 305);

controlling output of at least the audio data and the video data stored in the separate buffers such that at least the audio data and the video data stored in the separate buffers are associated with each other (*see, for example,* paragraph [0032] and Figure 3, steps 304 and 305); and

decoding, responsive to said controlling, at least the audio data and the video data stored in the separate buffers and outputting two or more decoded data (*see, for example,* paragraph [0035] and Figure 3, steps 309-311),

wherein, when a specific separate buffer becomes full of data (*see, for example,* paragraph [0031] and Figure 3, step 302):

stopping said distributing of at least the audio data and the video data into the specific separate buffer and said decoding of encoded data stored in the specific separate buffer (*see, for example,* paragraphs [0037]-[0039] and Figure 5);

initializing the specific separate buffer without initializing the buffer (*see, for example,* paragraphs [0037]-[0039] and Figure 5); and

resuming all processing which was stopped when the specific separate buffer became full after said initializing of the specific separate buffer (*see, for example,* paragraphs [0037]-[0039] and Figure 5).

Claim 8 recites a multiple decoding method for simultaneously decoding two or more encoded data from a broadcasting signal composed of a plurality of encoded data (*see, for example,* paragraphs [0001] and [0008] and Figure 1), the multiple decoding method comprising:

selecting a plurality of decoders for performing decoding and a plurality of separate buffers corresponding to the plurality of decoders, respectively, according to usage status of the plurality of decoders (*see, for example,* paragraphs [0029] and [0030] and Figure 2, steps 202-205);

extracting at least audio data and video data to be decoded and reproduced from the broadcasting signal (*see, for example,* paragraph 30 and Figure 2 steps 209 and 210);

storing at least the extracted audio data and video data in a buffer (*see, for example,* paragraph (*see, for example,* paragraph [0031] and Figure 3, step 303);

distributing at least the audio data and the video data stored in the buffer for each data type and respectively storing at least the audio data and the video data in the plurality of separate buffers according to each data type (*see, for example,* paragraph [0032] and Figure 3, steps 304 and 305);

controlling output of at least the audio data and the video data stored in the separate buffers such that at least the audio data and the video data stored in the separate buffers are associated with each other (*see, for example,* paragraph [0032] and Figure 3, steps 304 and 305); and

decoding, responsive to said controlling, at least the audio data and the video data stored in the separate buffers and outputting two or more decoded data (*see, for example,* paragraph [0035] and Figure 3, steps 309-311),

wherein, when a specific separate buffer becomes full of data (*see, for example,* paragraph [0031] and Figure 3, step 302):

discarding encoded data directed toward the specific separate buffer (*see, for example,* paragraphs [0040] and [0041] and Figure 6);

stopping said decoding of at least the audio data and the video data stored in the specific separate buffer (*see, for example,* paragraphs [0040] and [0041] and Figure 6);

initializing the specific separate buffer without initializing the buffer (*see, for example,* paragraph (*see, for example,* paragraphs [0040] and [0041] and Figure 6); and

resuming all processing which was stopped when the specific separate buffer became full after said initializing of the specific separate buffer, and releasing the discard of the encoded data (*see, for example,* paragraphs [0040] and [0041] and Figure 6).

Grounds of Rejection to be Reviewed on Appeal

I. Independent claims 3, 4, 7 and 8 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,332,058 (*hereinafter* the Kawakami reference) in view of U.S. Patent No. 5,159,447 (*hereinafter* the Siong reference) and U.S. Patent No. 5,159,447 (*hereinafter* the Haskell reference).

Arguments

I. Independent Claims 3, 4, 7 and 8 are Patentable Over the Kawakami Reference in view of the Siong Reference and the Haskell Reference

Initially, it is noted that the Federal Circuit in *In re Piasecki* stated that the U.S. Patent and Trademark Office has the burden under 35 U.S.C. §103 to establish a prima facie case of obviousness. (See *In re Piasecki*, 223 USPQ 785, 787 (Fed. Cir. 1984)). The Federal Circuit also has stated in *In re Fine* that the U.S. Patent and Trademark Office can only satisfy this burden by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. (See *In re Fine*, 5 USPQ2d 1596, 1598-99 (Fed. Cir. 1988)). As illustrated by these two court cases, the Examiner has the burden of establishing a prima facie case of obviousness by showing some objective teaching or knowledge generally available to one of ordinary skill in the art for combining the relevant portions of the references.

a. Re Claim 3

Claim 3 recites, in part:

(1) a data extractor for receiving a broadcasting signal and extracting at least audio data and video data, which are designated by control information, for storage in a buffer;

(2) a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in a plurality of separate buffers so as to be associated with each other in accordance with information for specifying the plurality of separate buffers and outputting, when a specific separate buffer becomes full of data, an overflow notification that the specific separate buffer overflows to a decoding controller; and

(3) a decoding controller that, upon reception of the overflow notification that the specific separate buffers overflows, instructs a data flow controller to stop data transfer to the specific separate buffer, instructs a decoder corresponding to the specific separate buffer to stop decoding, and instructs the separate buffer manager to initialize the specific separate buffer in the overflow state, while the buffer and the other separate buffers are not initialized and continue to operate in their normal manner.

The combination of the Kawakami reference, the Siong reference and the Haskell reference fails to disclose or suggest any of these features of claim 3.

The Kawakami reference discloses an MPEG server 16 that is adapted to receive an MPEG2 stream (information material) 14 including audio and video data. The information material 14 is generated by an MPEG encoder 12 that receives the audio and video data from a video tape recorder (VTR) 10. (See column 4, lines 47-60 and Figure 1).

The MPEG server 16 has an MPEG server core 18 that receives the information material 14 and an external controller 24 operable to supply a control signal 26 to the core 18. The MPEG server 16 also includes a number of hard disk drives (HDDs) 20, DMA buffers 30, a time-divisional multiplexing controller 40, gate controllers 32, decoder buffers 34 and decoders 22. (See column 4, line 47 – column 5, line 54 and Figures 1 and 2).

In a recordation operation, the MPEG server 16 receives a control signal 26 indicating that the MPEG server 16 is to record the information material 14. The MPEG server 16 then divides the information material 14 into a number of cells CE each having a size of four bytes. The cells CE are recorded on the HDDs 20 such that the first cell is stored on HDD 20-1, the second cell is stored on HDD 20-2, the third cell is stored on HDD 20-3, etc. Therefore, it is apparent that the information material 14 is split into a number of cells CE and the HDDs 20 are used to store the cells CE in parallel. (See column 5, line 10 – column 6, line 7).

In a reproduction operation, the MPEG server 16 receives a control signal 38 from a CPU group 36 indicating that the MPEG server 16 is to reproduce the information material 14. The cells CE are read from the HDDs 20 and stored in the DMA buffers 30 which respectively correspond to the HDDs 20. The cells CE are written to the DMA buffers 30 in clusters CT, which are larger than the cells CE. The controller 40 then controls the output of the information stored in the DMA buffers 30 such that desired information from each of the DMA buffers 30 is read at a desired time point. The gate controllers 32 operate so as to allow the information output by the DMA buffers 30 under the control of the controller 40 to only be supplied to the appropriate decoder buffer 34. (See column 5, lines 34-37; column 6, lines 46-49; column 7, lines 11-16 and 55-58; and Figures 1 and 2).

Once the information material 14 is properly stored in the decoder buffer 34, it is read out from the corresponding decoder 22 as packets PT and decoded into a video signal VS and an

audio signal AS to reproduce the information material 14. (See column 6, lines 42-59 and Figure 2).

In the rejection of claim 3, the MPEG server core 18 is relied upon as performing the operation of the claimed data extractor. However, as discussed above, the MPEG server core 18 receives the information material 14, which is an MPEG2 stream, from the MPEG encoder 12. The information material 14 is then stored directly on the HDDs 20. It is clear that the MPEG server core 18 does not perform any extraction of audio and video data (i.e., the information material 14) from a broadcast signal before storing the information material 14 on the HDDs 20 because the MPEG server core 18 only receives the audio and video data in the MPEG2 stream. Therefore, this feature is not disclosed or suggested by the Kawakami reference.

As exemplified in the above discussion of the data extractor of claim 3, it is apparent that the present invention, as recited in claim 3, is different from the MPEG sever 16 in the Kawakami reference because the broadcasting signal received by the data extractor has a data amount that cannot be controlled at the reception end (i.e., the claimed multiple decoding apparatus). Therefore, claim 3 recites the buffer and the plurality of separate buffers that are used to handle the data of the broadcasting signal. On the other hand, in the Kawakami reference, when a data overflow condition occurs, the data overflow condition is dealt with by directly controlling the data amount sent to the DMA buffers 30 by reducing or stopping the transmission of data from the HDDs 20. As a result, it is apparent that the Kawakami reference does not even contemplate the potential problem of handling a broadcasting signal addressed by the present invention, as recited in claim 3.

Regarding the separate buffer manager and the decoding controller of claim 3, it is apparent that the Kawakami reference fails to disclose or suggest these features, as admitted in the rejection. However, there is a suggestion in the rejection that the gate controllers 32 somehow relate to the claimed separate buffer manager because the gate controllers 32 are "for controlling the outputs of each of the respective plurality of separate buffers 34." Regarding this statement, as discussed above, the gate controllers 32 are only disclosed as controlling the writing of the information material 14 into the respective decoder buffers 34. (See column 5, lines 40-42). There is no disclosure or suggestion that the gate controllers 32 are capable of controlling the output of the decoder buffers 34. Further, based on the fact that the gate controllers 32 are illustrated in Figure 2 as being located upstream of the decoder buffers 34 and

do not appear to have the ability to issue a control signal to the decoder buffers 34, it is unclear how they would be capable of controlling the output of the decoder buffers 34.

In light of the above, in order for the combination of the Kawakami reference, the Siong reference and the Haskell reference to render claim 3 obvious, at least one of the Siong reference and the Haskell reference must disclose or suggest (1) the data extractor, (2) the separate buffer manager and (3) the decoding controller as recited in claim 3. Further, as mentioned above, it is necessary that the rejection set forth some objective teaching or knowledge generally available to one of ordinary skill in the art for combining the relevant portions of the references.

Regarding the Siong reference, it discloses a video decoding system including a micro-controller 6, a plurality of display buffers 7-9 and a corresponding plurality of decoding units 3-5, each including a multiplexed buffer 13 and a video decoder 14. The micro-controller 6 controls the decoding units 3-5 such that only one is outputting picture data to the corresponding display buffer 7-9 at a time. During the operation of the video decoding system, the micro-controller 6 monitors each of the multiplexed buffers 13 to determine when one of them contains enough data for performing decoding. Once the multiplexed buffer 13 contains enough data, the micro-controller 6 activates the associated video decoder 14 which decodes the data and outputs it to the display buffer 7-9. The video decoder 14 automatically stops decoding when the multiplexed buffer 13 is almost empty. Once the video decoder 14 stops decoding, it is necessary for the micro-controller 6 to activate the decoder 14 again. (See column 3, line 56 – column 4, line 27 and Figures 1 and 2).

In the rejection, the micro-controller 6 of the Siong reference is relied upon as disclosing the general concept of the claimed separate buffer manager. However, while the micro-controller 6 does indirectly control the output of the multiplexed buffers 13 by controlling when the corresponding video decoder 14 removes the data therefrom, it is apparent that the micro-controller 6 is in no way used in contemplation of buffer overflow and does not output an overflow notification. In fact, just the opposite is true, since the micro-controller 6 is concerned with whether or not the multiplexed buffer 13 has enough data for the corresponding video decoder 14, i.e., underflow.

The rejection indicates that one of ordinary skill in the art looking at the disclosure of the gate controllers 32 in the Kawakami reference and the disclosure of the micro-controller 6 in the Siong reference would reach the conclusion that the micro-controller 6 of the Siong reference

could be used in place of the gate controllers 32 to reduce the amount of hardware. However, this conclusion completely ignores the fact that neither the gate controllers 32, nor the micro-controller 6, operate in the same way as the claimed separate buffer manager. In other words, the gate controllers 32 and the micro-controller 6 do not individually or by combining their functions, output, when a specific separate buffer becomes full of data, an overflow notification that a specific separate buffer overflows to a decoding controller. As a result, the Siong reference does not address the deficiency of the Kawakami reference with respect to the separate buffer manager.

Further, it is apparent that the Siong reference also fails to disclose or suggest the claimed data extractor, since the video decoding system illustrated in Figure 1 receives a video bit stream that is immediately stored in a transmission buffer.

Additionally, as implied in the rejection, the Siong reference also fails to disclose or suggest the claimed decoding controller, since it fails to consider the occurrence of a buffer overflow. Therefore, in order for the combination to render claim 3 obvious, it is apparent that the Haskell reference must disclose or suggest (1) the data extractor, (2) the separate buffer manager and (3) the decoding controller, since the Kawakami reference and the Siong reference do not.

The Haskell reference discloses an encoding system including an encoder rate controller 103, an encoder 101 and a decoder buffer 205 (304). The encoder rate controller 103 monitors the fullness of the decoder buffer 205 (304) and should overflow of the decoder buffer 205 (304) be threatened, the encoder rate controller 103 controls the encoder 101 to reduce its output data rate. (See column 16, lines 7-27; column 17, line 66 – column 18, line 13; and Figures 1-4).

Further, the Haskell reference discloses that instead of the encoder rate controller 103 controlling the decoder buffer 205 (304) to reduce its output rate, other methods of overflow prevention can be utilized. For example, some of the packets of data can be deleted with the intention of sending them at a later time, or stuffing bits could be placed in an upstream multiplexer buffer 109 that would result in giving the decoder buffer 205 (304) time to empty. (See column 16, lines 27-39).

In the rejection, the above-mentioned sections of the Haskell reference are relied upon as providing support for the conclusion that the operations performed by the claimed decoding controller are somehow obvious. However, it is initially noted that all of the above sections of

the Haskell reference deal with the prevention of overflow of a buffer. On the other hand, the claimed decoding controller operates upon reception of an overflow notification that a specific separate buffer is full. In other words, the claimed decoding controller deals with overflow recovery after overflow has occurred, while the Haskell reference relates to overflow prevention. Therefore, the encoder rate controller 103 does not operate in the same manner as the claimed decoding controller.

More specifically, the encoder rate controller 103 does not, upon reception of an overflow notification that the specific separate buffers overflows, instruct a data flow controller to stop data transfer to the specific separate buffer, instruct a decoder corresponding to the specific separate buffer to stop decoding, and instruct a separate buffer manager to initialize the specific separate buffer in the overflow state, while the buffer and the other separate buffers are not initialized and continue to operate in their normal manner.

Since the Haskell reference clearly provides no disclosure or suggestion of the functions of the decoding controller, and does not even relate to overflow recovery, it is apparent that the comments related to how it would have been obvious to one of ordinary skill in the art based on the disclosure of the Haskell reference to initialize a buffer after overflow is without merit because the encoder rate controller 103 prevents overflow from even occurring. Therefore, it is clear that the Haskell reference fails to disclose or suggest the claimed decoding controller.

Further, it is apparent that the Haskell reference fails to disclose or suggest the claimed separate buffer manager which outputs an overflow notification that a specific buffer has overflowed, since the encoder rate controller 103 eliminates the possibility of buffer overflow.

Additionally, the Haskell reference also fails to disclose or suggest the claimed data extractor, since the encoding system illustrated in Figure 1 receives a video signal that is immediately encoded by a video encoder 10.

In light of the above discussion, it is apparent that none of the references either individually, or in combination, disclose or suggest (1) the data extractor, (2) the separate buffer manager, and (3) the decoding controller as recited in claim 3. Further, although the references were discussed one at a time for the sake of clarity, this should not be construed as attacking each of the references individually, since the discussion is in the context of how the combination of the references is relied upon in the rejection. As a result, claim 3 is patentable over the combination of the Kawakami reference, the Siong reference and the Haskell reference.

b. Re Claim 4

Claim 4 recites, in part:

(1) a data extractor for receiving a broadcasting signal and extracting at least audio data and video data, which are designated by control information, for storage in a buffer;

(2) a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in a plurality of separate buffers so as to be associated with each other in accordance with information for specifying the plurality of separate buffers and outputting, when a specific separate buffer becomes full of data, an overflow notification that the specific separate buffer overflows to a decoding controller; and

(3) a decoding controller that, upon reception of the overflow notification that the specific separate buffers overflows, instructs a data flow controller to discard encoded data directed toward the specific separate buffer, instructs a decoder corresponding to the specific separate buffer to stop decoding, and instructs the separate buffer manager to initialize the specific separate buffer, while the buffer and the other separate buffers are not initialized and continue to operate in their normal manner.

The combination of the Kawakami reference, the Siong reference and the Haskell reference fails to disclose or suggest any of these features of claim 4.

The Kawakami reference discloses an MPEG server 16 that is adapted to receive an MPEG2 stream (information material) 14 including audio and video data. The information material 14 is generated by an MPEG encoder 12 that receives the audio and video data from a video tape recorder (VTR) 10. (See column 4, lines 47-60 and Figure 1).

The MPEG server 16 has an MPEG server core 18 that receives the information material 14 and an external controller 24 operable to supply a control signal 26 to the core 18. The MPEG server 16 also includes a number of hard disk drives (HDDs) 20, DMA buffers 30, a time-divisional multiplexing controller 40, gate controllers 32, decoder buffers 34 and decoders 22. (See column 4, line 47 – column 5, line 54 and Figures 1 and 2).

In a recordation operation, the MPEG server 16 receives a control signal 26 indicating that the MPEG server 16 is to record the information material 14. The MPEG server 16 then divides the information material 14 into a number of cells CE each having a size of four bytes. The cells CE are recorded on the HDDs 20 such that the first cell is stored on HDD 20-1, the

second cell is stored on HDD 20-2, the third cell is stored on HDD 20-3, etc. Therefore, it is apparent that the information material 14 is split into a number of cells CE and the HDDs 20 are used to store the cells CE in parallel. (See column 5, line 10 – column 6, line 7).

In a reproduction operation, the MPEG server 16 receives a control signal 38 from a CPU group 36 indicating that the MPEG server 16 is to reproduce the information material 14. The cells CE are read from the HDDs 20 and stored in the DMA buffers 30 which respectively correspond to the HDDs 20. The cells CE are written to the DMA buffers 30 in clusters CT, which are larger than the cells CE. The controller 40 then controls the output of the information stored in the DMA buffers 30 such that desired information from each of the DMA buffers 30 is read at a desired time point. The gate controllers 32 operate so as to allow the information output by the DMA buffers 30 under the control of the controller 40 to only be supplied to the appropriate decoder buffer 34. (See column 5, lines 34-37; column 6, lines 46-49; column 7, lines 11-16 and 55-58; and Figures 1 and 2).

Once the information material 14 is properly stored in the decoder buffer 34, it is read out from the corresponding decoder 22 as packets PT and decoded into a video signal VS and an audio signal AS to reproduce the information material 14. (See column 6, lines 42-59 and Figure 2).

In the rejection of claim 4, the MPEG server core 18 is relied upon as performing the operation of the claimed data extractor. However, as discussed above, the MPEG server core 18 receives the information material 14, which is an MPEG2 stream, from the MPEG encoder 12. The information material 14 is then stored directly on the HDDs 20. It is clear that the MPEG server core 18 does not perform any extraction of audio and video data (i.e., the information material 14) from a broadcast signal before storing the information material 14 on the HDDs 20 because the MPEG server core 18 only receives the audio and video data in the MPEG2 stream. Therefore, this feature is not disclosed or suggested by the Kawakami reference.

As exemplified in the above discussion of the data extractor of claim 4, it is apparent that the present invention, as recited in claim 4, is different from the MPEG sever 16 in the Kawakami reference because the broadcasting signal received by the data extractor has a data amount that cannot be controlled at the reception end (i.e., the claimed multiple decoding apparatus). Therefore, claim 4 recites the buffer and the plurality of separate buffers that are used to handle the data of the broadcasting signal. On the other hand, in the Kawakami

reference, when a data overflow condition occurs, the data overflow condition is dealt with by directly controlling the data amount sent to the DMA buffers 30 by reducing or stopping the transmission of data from the HDDs 20. As a result, it is apparent that the Kawakami reference does not even contemplate the potential problem of handling a broadcasting signal addressed by the present invention, as recited in claim 4.

Regarding the separate buffer manager and the decoding controller of claim 4, it is apparent that the Kawakami reference fails to disclose or suggest these features, as admitted in the rejection. However, there is a suggestion in the rejection that the gate controllers 32 somehow relate to the claimed separate buffer manager because the gate controllers 32 are "for controlling the outputs of each of the respective plurality of separate buffers 34." Regarding this statement, as discussed above, the gate controllers 32 are only disclosed as controlling the writing of the information material 14 into the respective decoder buffers 34. (See column 5, lines 40-42). There is no disclosure or suggestion that the gate controllers 32 are capable of controlling the output of the decoder buffers 34. Further, based on the fact that the gate controllers 32 are illustrated in Figure 2 as being located upstream of the decoder buffers 34 and do not appear to have the ability to issue a control signal to the decoder buffers 34, it is unclear how they would be capable of controlling the output of the decoder buffers 34.

In light of the above, in order for the combination of the Kawakami reference, the Siong reference and the Haskell reference to render claim 4 obvious, at least one of the Siong reference and the Haskell reference must disclose or suggest (1) the data extractor, (2) the separate buffer manager and (3) the decoding controller as recited in claim 4. Further, as mentioned above, it is necessary that the rejection set forth some objective teaching or knowledge generally available to one of ordinary skill in the art for combining the relevant portions of the references.

Regarding the Siong reference, it discloses a video decoding system including a micro-controller 6, a plurality of display buffers 7-9 and a corresponding plurality of decoding units 3-5, each including a multiplexed buffer 13 and a video decoder 14. The micro-controller 6 controls the decoding units 3-5 such that only one is outputting picture data to the corresponding display buffer 7-9 at a time. During the operation of the video decoding system, the micro-controller 6 monitors each of the multiplexed buffers 13 to determine when one of them contains enough data for performing decoding. Once the multiplexed buffer 13 contains enough data, the micro-controller 6 activates the associated video decoder 14 which decodes the data and outputs

it to the display buffer 7-9. The video decoder 14 automatically stops decoding when the multiplexed buffer 13 is almost empty. Once the video decoder 14 stops decoding, it is necessary for the micro-controller 6 to activate the decoder 14 again. (See column 3, line 56 – column 4, line 27 and Figures 1 and 2).

In the rejection, the micro-controller 6 of the Siong reference is relied upon as disclosing the general concept of the claimed separate buffer manager. However, while the micro-controller 6 does indirectly control the output of the multiplexed buffers 13 by controlling when the corresponding video decoder 14 removes the data therefrom, it is apparent that the micro-controller 6 is in no way used in contemplation of buffer overflow and does not output an overflow notification. In fact, just the opposite is true, since the micro-controller 6 is concerned with whether or not the multiplexed buffer 13 has enough data for the corresponding video decoder 14, i.e., underflow.

The rejection indicates that one of ordinary skill in the art looking at the disclosure of the gate controllers 32 in the Kawakami reference and the disclosure of the micro-controller 6 in the Siong reference would reach the conclusion that the micro-controller 6 of the Siong reference could be used in place of the gate controllers 32 to reduce the amount of hardware. However, this conclusion completely ignores the fact that neither the gate controllers 32, nor the micro-controller 6, operate in the same way as the claimed separate buffer manager. In other words, the gate controllers 32 and the micro-controller 6 do not individually or by combining their functions, output, when a specific separate buffer becomes full of data, an overflow notification that a specific separate buffer overflows to a decoding controller. As a result, the Siong reference does not address the deficiency of the Kawakami reference with respect to the separate buffer manager.

Further, it is apparent that the Siong reference also fails to disclose or suggest the claimed data extractor, since the video decoding system illustrated in Figure 1 receives a video bit stream that is immediately stored in a transmission buffer.

Additionally, as implied in the rejection, the Siong reference also fails to disclose or suggest the claimed decoding controller, since it fails to consider the occurrence of a buffer overflow. Therefore, in order for the combination to render claim 4 obvious, it is apparent that the Haskell reference must disclose or suggest (1) the data extractor, (2) the separate buffer

manager and (3) the decoding controller, since the Kawakami reference and the Siong reference do not.

The Haskell reference discloses an encoding system including an encoder rate controller 103, an encoder 101 and a decoder buffer 205 (304). The encoder rate controller 103 monitors the fullness of the decoder buffer 205 (304) and should overflow of the decoder buffer 205 (304) be threatened, the encoder rate controller 103 controls the encoder 101 to reduce its output data rate. (See column 16, lines 7-27; column 17, line 66 – column 18, line 13; and Figures 1-4).

Further, the Haskell reference discloses that instead of the encoder rate controller 103 controlling the decoder buffer 205 (304) to reduce its output rate, other methods of overflow prevention can be utilized. For example, some of the packets of data can be deleted with the intention of sending them at a later time, or stuffing bits could be placed in an upstream multiplexer buffer 109 that would result in giving the decoder buffer 205 (304) time to empty. (See column 16, lines 27-39).

In the rejection, the above-mentioned sections of the Haskell reference are relied upon as providing support for the conclusion that the operations performed by the claimed decoding controller are obvious. Regarding this, it is noted that the Haskell reference does disclose the deletion of data packets which is similar to discarding of data as recited with regard to the decoding controller. However, it is noted that all of the above sections of the Haskell reference deal with the prevention of overflow of a buffer. On the other hand, the claimed decoding controller operates upon reception of an overflow notification that a specific separate buffer is full. In other words, the claimed decoding controller deals with overflow recovery after overflow has occurred, while the Haskell reference relates to overflow prevention. Therefore, the encoder rate controller 103 does not operate in the same manner as the claimed decoding controller.

More specifically, the encoder rate controller 103 does not, upon reception of an overflow notification that the specific separate buffers overflows, instruct a data flow controller to discard encoded data directed toward the specific separate buffer, instruct a decoder corresponding to the specific separate buffer to stop decoding, and instruct the separate buffer manager to initialize the specific separate buffer, while the buffer and the other separate buffers are not initialized and continue to operate in their normal manner.

Since the Haskell reference clearly provides no disclosure or suggestion of the functions of the decoding controller, and does not even relate to overflow recovery, it is apparent that the comments related to how it would have been obvious to one of ordinary skill in the art based on the disclosure of the Haskell reference to initialize a buffer after overflow is without merit because the encoder rate controller 103 prevents overflow from even occurring. Therefore, it is clear that the Haskell reference fails to disclose or suggest the claimed decoding controller.

Further, it is apparent that the Haskell reference fails to disclose or suggest the claimed separate buffer manager which outputs an overflow notification that a specific buffer has overflowed, since the encoder rate controller 103 eliminates the possibility of buffer overflow.

Additionally, the Haskell reference also fails to disclose or suggest the claimed data extractor, since the encoding system illustrated in Figure 1 receives a video signal that is immediately encoded by a video encoder 10.

In light of the above discussion, it is apparent that none of the references either individually, or in combination, disclose or suggest (1) the data extractor, (2) the separate buffer manager, and (3) the decoding controller as recited in claim 4. Further, although the references were discussed one at a time for the sake of clarity, this should not be construed as attacking each of the references individually, since the discussion is in the context of how the combination of the references is relied upon in the rejection. As a result, claim 4 is patentable over the combination of the Kawakami reference, the Siong reference and the Haskell reference.

c. Re Claim 7

Claim 7 recites, in part:

extracting at least audio data and video data to be decoded and reproduced from a broadcasting signal, for storage in a buffer,

wherein, when a specific separate buffer becomes full of data:

stopping distributing of at least the audio data and the video data into the specific separate buffer and decoding of encoded data stored in the specific separate buffer;

initializing the specific separate buffer without initializing the buffer; and

resuming all processing which was stopped when the specific separate buffer became full after the initializing of the specific separate buffer.

Therefore, it is apparent that claim 7 is patentable over the combination of the Kawakami reference, the Siong reference and the Haskell reference for reasons similar to those set forth above regarding the data extractor and the decoding controller of claim 3.

d. Re Claim 8

extracting at least audio data and video data to be decoded and reproduced from a broadcasting signal, for storage in a buffer,

wherein, when a specific separate buffer becomes full of data:

discarding encoded data directed toward the specific separate buffer;

stopping decoding of at least the audio data and the video data stored in the specific separate buffer;

initializing the specific separate buffer without initializing the buffer; and

resuming all processing which was stopped when the specific separate buffer became full after the initializing of the specific separate buffer, and releasing the discard of the encoded data.

Therefore, it is apparent that claim 8 is patentable over the combination of the Kawakami reference, the Siong reference and the Haskell reference for reasons similar to those set forth above regarding the data extractor and the decoding controller of claim 4.

Conclusion

In view of the above, it is respectfully submitted that independent claims 3, 4, 7 and 8 are patentable over the Kawakami reference in view of the Siong reference and the Haskell reference. Therefore, rejected claims 3, 4, 7 and 8 are allowable. Accordingly, the Board is requested to reverse the rejections set forth in the final Office Action of October 24, 2006.

This brief is submitted with the requisite fee of $500.00.

Respectfully submitted,

Akira KAMIYA

By: _____
David M. Ovedovitz
Registration No. 45,336
Attorney for Applicant

DMO/jmj
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
March 20, 2007

APPENDIX I - Claims on Appeal

3. A multiple decoding apparatus for receiving a broadcasting signal composed of a plurality of encoded data and for simultaneously decoding two or more of the encoded data, the multiple decoding apparatus comprising:

a reproduction controller for outputting control information related to decoding and reproduction of data;

a data extractor for receiving the broadcasting signal and extracting at least audio data and video data which are designated by the control information;

a buffer for storing at least the audio data and the video data extracted by said data extractor;

a buffer manager for controlling said buffer in accordance with the control information for said buffer;

a data flow controller for distributing at least the audio data and the video data stored in said buffer for each data type and transferring at least the audio data and the video data in accordance with provided transfer conditions;

a plurality of separate buffers for respectively storing at least the audio data and the video data distributed and transferred by said data flow controller according to each data type;

a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in said plurality of separate buffers so as to be associated with each other in accordance with information for specifying said plurality of separate buffers;

a plurality of decoders respectively corresponding to said plurality of separate buffers for decoding at least the audio data and the video data stored in said plurality of separate buffers and outputting two or more decoded data; and

a decoding controller for selecting a separate buffer and a decoder, which are used for the decoding, according to a usage status of said decoder from among said plurality of separate buffers and said plurality of decoders in accordance with the control information, and outputting information related to said separate buffer selected by said decoding controller, the transfer conditions based on said separate buffer selected by said decoding controller, and an instruction to start decoding, respectively, to said separate buffer manager, said data flow controller, and said decoder selected by said decoding controller, wherein

said separate buffer manager outputs, when a specific separate buffer becomes full of data, an overflow notification that said specific separate buffer overflows to said decoding controller,

said decoding controller outputs, upon receipt of the overflow notification that said specific separate buffer overflows, an instruction to stop data transfer to said specific separate buffer to said data flow controller, outputs an instruction to stop decoding to a decoder corresponding to said specific separate buffer, and outputs an instruction to initialize said specific separate buffer to said separate buffer manager,

said separate buffer manager initializes said specific separate buffer in accordance with the instruction to initialize said specific separate buffer from said decoding controller without initializing said buffer, and

the multiple decoding apparatus resumes all processing which was stopped as a result of said specific separate buffer becoming full after said specific separate buffer is initialized.

4. A multiple decoding apparatus for receiving a broadcasting signal composed of a plurality of encoded data and for simultaneously decoding two or more of the encoded data, the multiple decoding apparatus comprising:

a reproduction controller for outputting control information related to decoding and reproduction of data;

a data extractor for receiving the broadcasting signal and extracting at least audio data and video data which are designated by the control information;

a buffer for storing at least the audio data and the video data extracted by said data extractor;

a buffer manager for controlling said buffer in accordance with the control information for said buffer;

a data flow controller for distributing at least the audio data and the video data stored in said buffer for each data type and transferring at least the audio data and the video data in accordance with provided transfer conditions;

a plurality of separate buffers for respectively storing at least the audio data and the video data distributed and transferred by said data flow controller according to each data type;

a separate buffer manager for controlling output of at least the audio data and the video data respectively stored in said plurality of separate buffers so as to be associated with each other in accordance with information for specifying said plurality of separate buffers;

a plurality of decoders respectively corresponding to said plurality of separate buffers for decoding at least the audio data and the video data stored in said plurality of separate buffers and outputting two or more decoded data; and

a decoding controller for selecting a separate buffer and a decoder, which are used for the decoding, according to a usage status of said decoder from among said plurality of separate buffers and said plurality of decoders in accordance with the control information, and outputting information related to said separate buffer selected by said decoding controller, the transfer conditions based on said separate buffer selected by said decoding controller, and an instruction to start decoding, respectively, to said separate buffer manager, said data flow controller, and said decoder selected by said decoding controller, wherein

said separate buffer manager outputs, when a specific separate buffer becomes full of data, an overflow notification that said specific separate buffer overflows to said decoding controller,

said decoding controller outputs, upon receipt of the overflow notification that said specific separate buffer overflows, an instruction to discard encoded data directed toward said specific separate buffer to said data flow controller, outputs an instruction to stop decoding to a decoder corresponding to said specific separate buffer, and outputs an instruction to initialize said specific separate buffer to said separate buffer manager,

said separate buffer manager initializes said specific separate buffer in accordance with the instruction to initialize said specific separate buffer from said decoding controller without initializing said buffer, and

the multiple decoding apparatus resumes all processing which was stopped as a result of said specific separate buffer becoming full, and the discard of the encoded data is released after said specific separate buffer is initialized.

7. A multiple decoding method for simultaneously decoding two or more encoded data from a broadcasting signal composed of a plurality of encoded data, the multiple decoding method comprising:

selecting a plurality of decoders for performing decoding and a plurality of separate buffers corresponding to the plurality of decoders, respectively, according to usage status of the plurality of decoders;

extracting at least audio data and video data to be decoded and reproduced from the broadcasting signal;

storing at least the extracted audio data and video data in a buffer;

distributing at least the audio data and the video data stored in the buffer for each data type and respectively storing at least the audio data and the video data in the plurality of separate buffers according to each data type;

controlling output of at least the audio data and the video data stored in the separate buffers such that at least the audio data and the video data stored in the separate buffers are associated with each other; and

decoding, responsive to said controlling, at least the audio data and the video data stored in the separate buffers and outputting two or more decoded data,

wherein, when a specific separate buffer becomes full of data:

stopping said distributing of at least the audio data and the video data into the specific separate buffer and said decoding of encoded data stored in the specific separate buffer;

initializing the specific separate buffer without initializing the buffer; and

resuming all processing which was stopped when the specific separate buffer became full after said initializing of the specific separate buffer.

8. A multiple decoding method for simultaneously decoding two or more encoded data from a broadcasting signal composed of a plurality of encoded data, the multiple decoding method comprising:

selecting a plurality of decoders for performing decoding and a plurality of separate buffers corresponding to the plurality of decoders, respectively, according to usage status of the plurality of decoders;

extracting at least audio data and video data to be decoded and reproduced from the broadcasting signal;

storing at least the extracted audio data and video data in a buffer;

distributing at least the audio data and the video data stored in the buffer for each data type and respectively storing at least the audio data and the video data in the plurality of separate buffers according to each data type;

controlling output of at least the audio data and the video data stored in the separate buffers such that at least the audio data and the video data stored in the separate buffers are associated with each other; and

decoding, responsive to said controlling, at least the audio data and the video data stored in the separate buffers and outputting two or more decoded data,

wherein, when a specific separate buffer becomes full of data:

discarding encoded data directed toward the specific separate buffer;

stopping said decoding of at least the audio data and the video data stored in the specific separate buffer;

initializing the specific separate buffer without initializing the buffer; and

resuming all processing which was stopped when the specific separate buffer became full after said initializing of the specific separate buffer, and releasing the discard of the encoded data.

APPENDIX II - Evidence

There is no evidence relied on by Appellant.

APPENDIX III - Related Proceedings

As indicated above, there are no related appeals or interferences.